XP-002328425

Intelligent Paper

Marc Dymetman and Max Copperman

Xerox Research Centre Europe
6, chemin de Maupertuis
38240 Meylan, France

{dymetman, copperman}@xrce.xerox.com

Intelligere: Latin for 'understand', 'appreciate';
from *inter:* 'between', 'among' and *legere:* 'to pick', 'to gather', 'to choose', 'to read'.
Intelligent: Who can read between the lines.

1 Introduction

Some researchers have recently been calling attention to certain lasting "affordances" of paper documents over digital ones [21, 17]. Whereas the relationship between the two media is often assumed to be one of competition, in fact it is one of complementarity. While digital documents are *dynamic* (evolving in time), *immaterial* (made of informational substance), and *agentive* (capable of initiating actions), paper documents are:

Permanent: once printed they are "frozen", provide a stable memory of a document historical state, can be destroyed but not altered;
Portable: they are light, can be transported autonomously, can be transferred hand-to-hand in a social encounter;
Contextualized: they are parts of the physical world, have a material richness and re-markability, their varied appearance and physical surroundings ensure an abundant mnemonic context.

Intelligent Paper is a proposal for bridging the gap between the physical world and the digital world. Intelligent Paper is made up of three elements:

Support Each sheet of paper (physical page) is identified by a code *page-id* which characterizes it uniquely among all pages world-wide;
Input device An input device, called a pointer, is provided. When the user positions the pointer on the page and clicks it, the pointer's coordinates *pointer-loc* are recognized;
Communication infrastructure The pair <*page-id, pointer-loc*> is sent over the Web; the *page-id* is decoded as a network address, and the *pointer-loc* is interpreted by a program at this address. This results in an appropriate output action on an output peripheral close to the user.

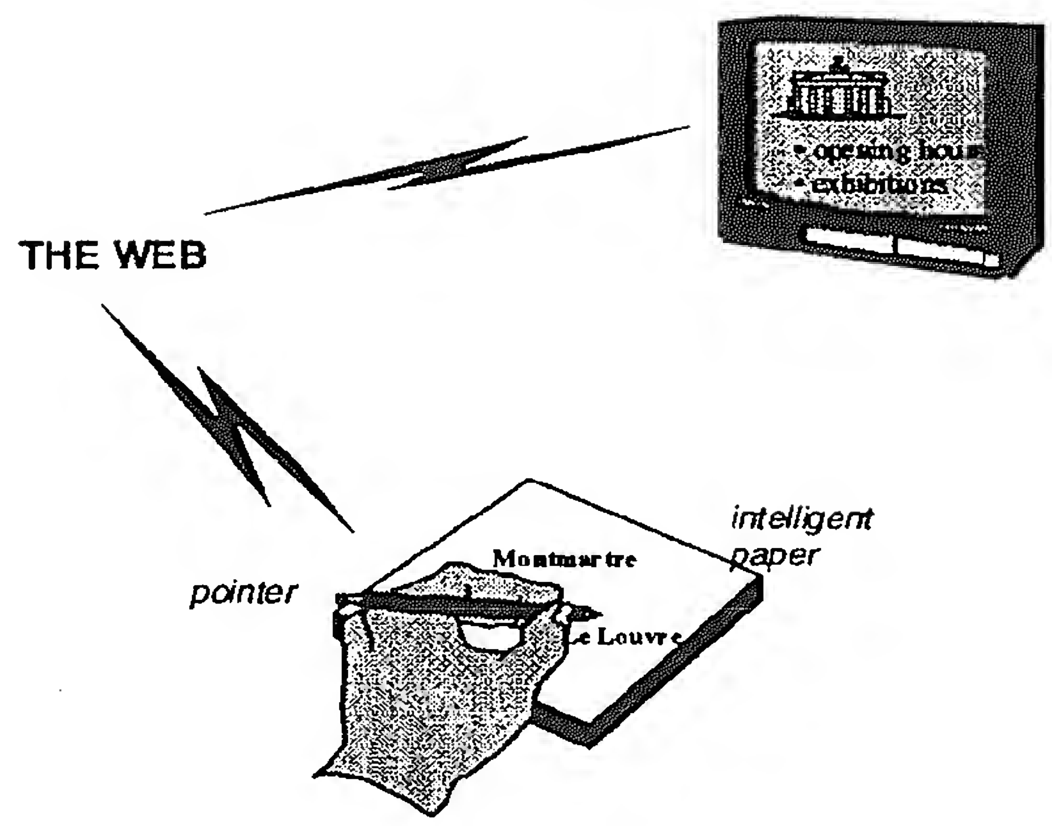

Fig. 1. Intelligent Paper.

An illustration is given in Fig. 1. The user is looking at a paper map of Paris, indicating such places of interest as the Louvre or the Sacré-Coeur. She positions the pointer over the Louvre, and clicks. The *page-id* is decoded by the communication infrastructure into a URL belonging to the publisher of the map. This publisher owns an electronic version of the map, which allows it to realize that the transmitted *pointer-loc* corresponds to the location of certain marks on the page, representing "Le Louvre". The publisher associates freely chosen actions with these marks, such as sending a video presentation of the museum to a TV peripheral close-by.

With Intelligent Paper, each piece of paper retains all conventional stationery qualities. A book, a newspaper, a prospectus can still be used in the usual ways, without any special equipment. However, if the user owns a pointer, and is near a peripheral device connected to the Web, the paper document assumes the behavior of a touch-sensitive screen. It becomes enriched with arbitrarily complex information, provided by the document's publisher at its web site. How is this achieved?

Intelligent Paper pages are standard sheets of paper entirely covered with printed marks, invisible to the human eye, but visible to the pointer. These marks convey two types of information: unique page identification and local coordinates relative to the page frame (plus possibly an encryption code, see section 5). When the user positions the pointer device at a certain location on the page, the pointer reads these invisible marks and determines from them the unique page identification and its own current location on the page, and then sends this information for interpretation over the web.

These sheets are produced by publishers, who buy apparently blank sheets of Intelligent Paper from an authorized producer. The publishers can mark them with conventional visible inks in any way they choose.[1] For each sheet of Intelligent Paper, the infrastructure provided by the Intelligent Paper producer will ensure that, when the end-user clicks the pointer, the *<page-id, pointer-loc>* pair is routed to the publisher, who is then responsible for associating whatever actions he chooses with this pair. These actions depend on the internal representation — called the digital page — of the physical page, stored at the publisher's site. This digital-page contains descriptions of actions to be associated with certain areas on the physical page. It may also contain a full digital counterpart of the visible marks on the physical page. *In this way, the physical page behaves as a "window" displaying the visual content of its digital counterpart, and the pointer behaves as a "mouse" moving over this window, activating links, selecting content and performing actions.* The only difference between this and the standard computer screen and mouse is that the physical page cannot modify its display state.

There is currently a profusion of proposals for linking paper documents to the electronic world:

Over-desk video Some researchers have proposed mounting a video-camera above the user's desk and monitoring the user's gestures when he points or writes on paper documents ("digital desk") [15, 24, 19];

Memory notes Others have discussed ways in which written annotations on paper can be made during an audio or video recording and later used as fast indexes into the recording [23] (see also [14] for a related approach);

Embedded data Products have been released which permit the embedding of digital data in printed documents (SmartPaper™[12]);[2]

Marking active paper areas The PaperLink project [3] augments paper documents with electronic features by permitting the user to make marks on paper using a highlighter pen equipped with a camera; when the user later uses the pen to pick up these regions, pattern-recognition techniques allow the system to execute commands associated with these marks.

Dynamic paper Physical and chemical extensions of paper-like supports aim to make paper capable of changing its state in time, either for displaying new information [22, 11], or for recording data which can be recovered by reading devices [25, 2].

Intelligent Paper is distinguished among the approaches to connecting the paper world to the digital world by the view it takes on two points:

[1] There may be some constraints on the types of inks used, see section 3. Also, the publishers may have the capabilities of producing the invisible substrate themselves.

[2] SmartPaper uses DataGlyphs, as Intelligent Paper does, but in a quite different way. It is based on full-page scanning. There is no notion of the position of a pointer, nor is there a notion of a generic encoding substrate (Intelligent Paper's encoding of *page-id* and *pointer-loc*). Mostly, data are encoded directly into the paper, rather than being *addressed* through double indirection (see section 3).

1. The paper-pointer relationship is identical to the usual screen-mouse relationship;[3]
2. Each physical-page is tightly coupled to its digital-page counterpart; the connection between the pair (physical-page, digital-page) being maintained across arbitrary distances through the globally unique *page-id* and the infrastructure routing scheme.

The remainder of this paper is organized as follows. Section 2 illustrates some of the uses of Intelligent Paper by discussing the scenario of a student reading a multimedia-augmented "Intelligent Book" publication of the play "Othello". Section 3 presents the underlying technology of Intelligent Paper. Section 4 shows further application examples. Section 5 introduces a few extensions and variants of the primary concept.

2 An example: Reading Othello

Ariane, a French student of English literature, is doing some research for an essay on Shakespeare's drama, Othello. Ariane is in bed, in front of her web-addressable TV screen. She is holding a copy of Othello, printed on Intelligent Paper by the "Literature Classics" publishing company.

Here are some of the actions Ariane can carry out:

Triggering output actions Ariane is in doubt about the meaning of a certain word. She directs the pointer to this word and clicks; an explanatory note for the word is displayed on the screen. She would like the current line of dialogue to be pronounced by an actor for her; she first directs the pointer to a certain square on the top of the page labelled RECITE; then she directs the pointer to the beginning of the line she is interested in; the line is recited by Laurence Olivier. She now goes to the beginning of the next scene, points to the box PLAY SCENE, and Orson Welles' interpretation of the scene is performed for her.

Using the pointer as a mouse Ariane now decides to have a look at the main Web page for this edition of Othello. She clicks on the box MAIN WEB PAGE on the top of the physical page. The Web page appears on the screen. Ariane points on the box MOUSE MODE and can now use the pointer as a mouse, with the page as the mouse-pad: moving the pointer on the page results in relative movements of the arrow on the screen; pointer clicks are treated as mouse clicks. In this way she can move through Web pages in the usual way. At one point she finds an interesting description of "Venice and the battle of Lepanto".

Selecting content Returning to reading the dialogue, Ariane selects the word "exsufflicate" that she would like to have translated into French. She clicks

[3] David Hecht [8] has independently proposed the same idea. Peter Robinson [20] also proposes, in a "digital desk" context, a closely related approach.

on the box TRANSLATIONS on the top of the physical page. A special menu appears on the screen, giving a choice of several languages. She selects French. She now has a choice between either a word-in-context automatic translation or a number of standard literary translations. She selects Gide's translation, then directs the pointer to the difficult word. Gide's translation of the line appears on the screen, with the word(s) corresponding to "exsufflicate" singled out in boldface.

Annotating Ariane is now reading a passage in the play which reminds her of the Web page on the naval battle of Lepanto. She uses a normal pencil to write a small note to herself: "Lepanto". She goes back to the Web page and selects the box LINK TO WEB PAGE on the paper page. She then moves the pointer to the penciled note she just made, and clicks. A link is now registered between the location on the physical page where the note is penciled and the current Web page. When, on a later sitting, Ariane sees her margin note, she will be able to click on it with the pointer and the Web page "Venice and the battle of Lepanto" will pop on the screen.

This example illustrates some of the uses of Intelligent Paper. Before investigating other applications, we will now turn our attention to the technology underlying Intelligent Paper.

3 Technology

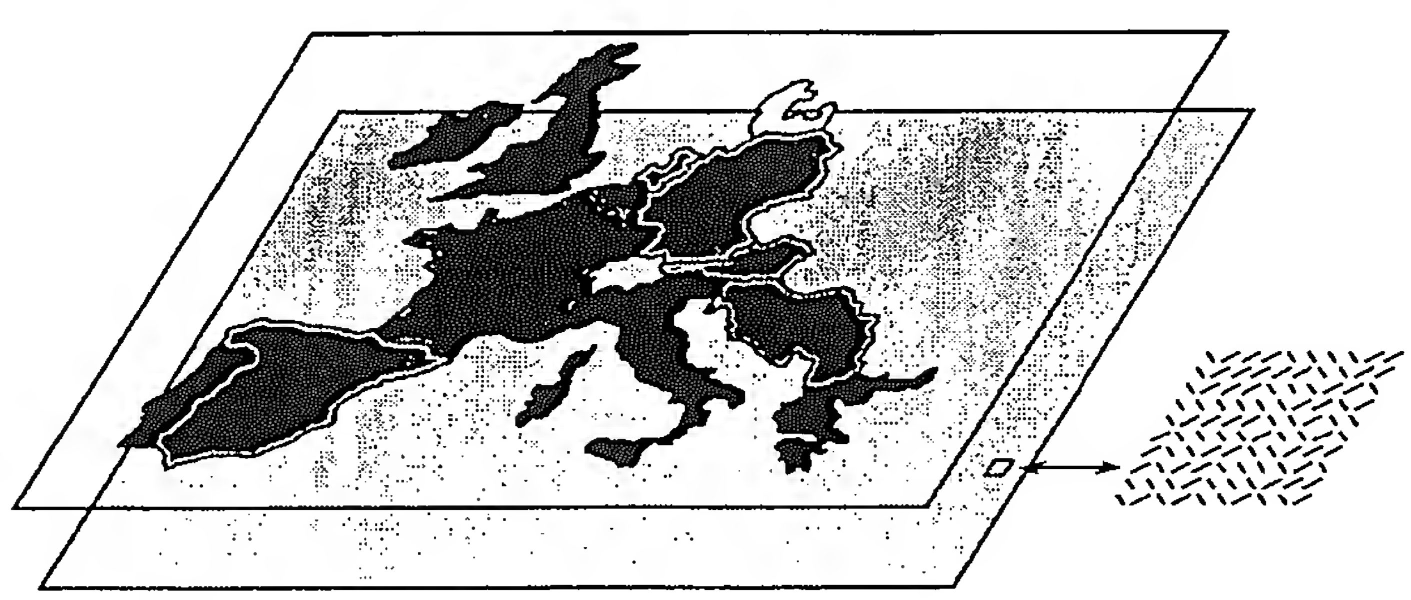

Fig. 2. A map of Europe printed on Intelligent Paper. Two layers of ink are printed on the paper support. The first layer, shown in gray, is the code layer. It encodes *page-id* and *pointer-loc* information via marks printed in invisible ink. A blow-up of these marks is shown on the right, where Xerox DataGlyphs™ have been used as the encoding mechanism. The second layer is printed in conventional coloured inks and is visible to the user.

Code layer The basic idea is illustrated in Fig. 2. DataGlyphs™ is a technology developed by Xerox for encoding digital information via printed marks on paper [7]. With the current DataGlyphs technology, it is possible to encode on the order of 400 bytes of data per square inch, using a 300 dpi printer.[4] DataGlyphs come in several variants. We will consider here one of the simplest variants, in which square cells about a quarter inch in length are laid close to each other. Each such cell contains on the order of 150 bits of data. We will assume that each cell on the page can be identified by a pair of integers (i, j), with $0 \leq i, j < 2^8$ (accommodating pages up to 5 feet (1.6 m) square).

Imagine now a small pen-shaped camera — the pointer — which, when in contact with the paper, is able to view such a cell in full. Suppose that, of the 150 bits of available data in the cell, 16 bits are reserved for coding the identifier (i, j) for the cell. The pointer can use these numbers to identify the cell, and it can also use the *position of the cell image relative to its field of view* to detect its own position relative to the cell. Therefore, it can detect its own precise position *pointer-loc* $= (x, y)$ relative to the physical page (where x, y are floating-point numbers).[5]

Once provision has been made for detecting the pointer's position relative to the page, there still remain on the order of 130 bits of data that are decoded by the pointer. A subset[6] of these bits is the page-identifier *page-id* used for identifying the digital-page counterpart of the physical page.

Page identifiers and the first indirection level If every inhabitant of the earth was to produce 80 thousands sheets of Intelligent Paper a day for the next century, then a *page-id* 64 bits in length would be enough to uniquely identify all the digital-pages needed.

Thus, a huge number of digital documents anywhere in the world can be uniquely referenced by a 64 bit *page-id*: the physical pages serve as global *indexes* to their digital-page counterparts. How is the mapping of the *page-id* onto the digital-page done in reality?

The most efficient way — in terms of *page-id* length, which is costly in terms of paper area used — of performing the mapping is by using a *router* which, given a *page-id*, returns the full URL of the digital page referenced by this *page-id*. In this manner, a *page-id* of length n bits can effectively reference 2^n different digital-pages. Such a router, however, can require huge tables for storing the *page-id*-URL relation. This problem can be approached in at least two ways: (i) Intelligent Paper pages can be sold at a price that covers the cost of storing the *page-id*-URL relation; (ii) the decoding of the address can be done in two (or

[4] This figure does not take into account error-correction, which can be parametrized in the DataGlyph technology. Depending on the level of error-correction chosen, the amount of usable information can be somewhat lower than the figure given.

[5] More sophisticated DataGlyph schemes for detecting the pointer's position using a small number of bits exist which do not require the camera to see a full cell, but that allow a maximal use of whatever area is captured by the camera [9].

[6] We will see uses for the remaining bits in section 5.

more) steps: the router decodes some prefix of the *page-id*— this prefix does not need to be of fixed length, but the length can vary depending on the actual *page-id* decoded — and then passes the remaining bits to a second router, typically one at the publisher's site. This way of proceeding exploits the natural tendency of publisher to buy Intelligent Paper sheets in bulk, so that it may be known by the first router that a certain number of consecutive *page-ids* are "owned" by a certain publisher.

Coding a *page-id* on paper is more efficient than coding full URLs directly; these need to be human-readable and mnemonic, and so use more space than is necessary for unique identification: a URL can easily reach 40 characters in length, a compared to the 8 bytes needed for a 64 bit *page-id*; even more seriously, the length of the URL cannot be bounded *a priori*.

Digital pages and the second indirection level The *page-id* and the router provide a *first indirection level* when interpreting the user's pointing actions. Once the digital page has been identified thanks to this first indirection, a *second level of indirection* is provided by the *pointer-loc*.[7]

When the user clicks at a certain location (x, y) on the page, and sends these coordinates to the digital-page identified in the first step, the pair (x, y) may serve as an index into the actions associated with this digital page, whether they are output actions such as displaying information on the user's peripheral, content selecting actions (in a given system state, the user's pointing may be interpreted as "Start selecting text", in the next state as "End selecting text"), or any other kind.

The mapping of positions to actions is the responsibility of the digital page and can be done in several ways. The problem is similar to the situation of interpreting mouse clicks in an hypertext document displayed on a conventional screen. Indeed, Intelligent Paper could be said to be an hypertext document displayed on a passive paper screen. It is therefore tempting to import as much of the existing GUI technology from the traditional "active display" world to the "passive display" world of Intelligent Paper. The approach that we are currently investigating is to use the Adobe Acrobat™ suite of products [1]. In particular, Acrobat Exchange permits one to display documents containing hyperlinks on a screen. These documents can be produced from arbitrrary PostScript files and have the same look and disposition on the screen as the printed PostScript documents have on paper. Because of this isomorphism between the screen and the paper version, it is relatively simple to map a position of the paper document into a position on the screen. Pointing actions on the paper can therefore be simulated by virtual mouse actions on the display; every action that the user could have performed on the screen, he can perform on the paper using this simulation.

Pointer Several technologies exist for the pointer's camera. They range from high-speed, high-resolution, CCD cameras to low-end hand-held scanners. The

[7] A similar idea to these two indirection levels appears in [20].

cost of these devices is constantly decreasing. Also decreasing is the cost of wireless telephony, so that it will shortly become realistic to tether the pointer to an Internet-enabled phone; it may even be possible in the near future to include wireless communications capabilities *inside* the pointer.

Visible and invisible inks In the description so far, we have assumed that the code-layer is printed using invisible ink. There is a technological challenge here. Several solutions exist; some of which use infrared inks, other ultra-violet ones. Also, in the scenario we have been describing, the code layer is printed under the visible ink layer; it is also possible to print the invisible ink layer over the visible one. Each choice has advantages and disadvantages. Printing the invisible ink layer under the visible one permits to distribute white sheets of Intelligent Paper which can then be printed (or written above) in visible inks in the traditional manner, but there is the challenge of recovering the invisible code through the layer of visible inks. These issues are related to questions about the light-absorption spectra of both visible and invisible inks.

Color printing uses three basic colored inks: cyan, magenta and yellow (sometimes black ink is also used but we will ignore that here). These inks work as translucid filters which each eliminate a certain range of frequencies in the white light spectrum. By so-called "subtractive" combination of colors, an arbitrary color can be reproduced by superposing the three basic colors in proper quantities [4]. In an idealized situation, each ink removes a strictly delimited interval of visible range frequencies and leaves untouched the other frequencies. One can in principle add a new "invisible" ink layer which removes frequencies only in a specific interval of invisible frequencies. This layer does not, in principle, interfere with the three visible layers, whether it is printed under or over them, and it can be detected by a camera equipped with an adequate optical filter. In practice, most commercial colour inks do not have a perfect absorption spectrum, and finding the right chemical composition for invisible inks is an active research area [5, 6].

If one does not wish to use invisible ink, partial implementations of Intelligent Paper are possible. Because visible DataGlyphs are minimally obstructive visually, they can be used as a gray background in otherwise white areas of paper. In this way, at least some areas of the paper document can be made active (for instance areas under words in running text), and many of the capabilities of Intelligent Paper retained.

4 Some Applications

Getting information

Maps A map of Paris can provide electronic information about public transportation; by selecting two different points on the map, the user can be told about the most efficient bus and subway routes. Events at a certain location, such as the Opéra, can be updated in real time on the Web site, though the user

still uses the same old physical copy of the map. One single, static, paper document becomes dynamic by being linked to constantly updated digital data.

Pocket encyclopedia A large encyclopedia can be condensed into a pocket version, giving concise descriptions, while the complete version is accessed through the Web or on a CD. A large dictionary can be produced on paper in a reduced format, while retaining the capabilities of the full version, and having extended functionalities such as thesaurus searches.

Periodicals A travel magazine can be augmented with touristic information about places described. A music journal can be enriched with recordings delivered either through the Web or on a CD distributed with the magazine. A scientific journal can have live bibliographic links permitting direct access to the text of cited articles. A professional newsletter can allow access to financial databases.

Daily papers In this age of immediate information, dailies have difficulties competing with media such as TV and radio. By being printed on Intelligent Paper, they can retain their specific advantages —journalistic outlook, in-depth analysis — while allowing instant information updates. Pointing on the headline for "French socialists estimated to win by narrow margin", the reader can have instant access to the latest polls, which can be delivered to her on a TV screen, on a computer, or through the telephone. Other functionalities may include pointing at a name to find all its mentions in recent issues or to display a biography, navigating to related recent articles, or selecting a TV broadcast by clicking on its entry in a program listing.

Active personal notebooks A small personal address book may contain in handwritten form frequently accessed information, such as name or telephone number, while less frequently accessed information, such as street address, photograph, or personal notes, are relegated to digital form. Personal handwritten annotations in a notebook can link to web sites, to personal documents, or even be associated with actions such as "start the coffee machine".

Meeting agenda Intelligent Paper sheets with a printed agenda can be distributed to participants in a meeting. These sheets can be used later by them to access recorded minutes of the agenda items, to find information about other meeting attendants, or to consult status of follow-up actions.

Interacting with services

Hotel reservations They can be made by accessing hotel sites from a tourist guide or a magazine. When the hotel page appears, the user may enter textual information (his name, address, etc.) through a small virtual keyboard on the screen. Alternatively, this keyboard can be printed directly on the physical

page. In this case, access to a screen peripheral is not necessary: the user could type his phone number on paper, and voice instructions and feedback could be given to him over the phone.[8]

Product catalogues Furniture, music or video catalogues can be produced on paper and the orders performed through the Web. An Intelligent Paper catalog sent by standard mail to a potential customer can be uniquely identified through the *page-id*, and so the user's address and ordering record is known at the time the pointer is clicked over an item.

Forms Examination forms can be printed on paper, and the student can fill them by pointing at locations on the form. The examination data become immediately available to the examiners.

5 Extensions and Variants

We finally examine a few extensions and variants of the Intelligent Paper idea.

Writer-Pointer A conceptually simple, but consequential, extension of the pointer device consists in providing it with a conventional writing tip (ball-point, pencil, ...). When the user writes on a white Intelligent Paper sheet using this *writer-pointer*, the position of the tip is monitored by the pointer camera in real-time. The marks can then be recorded in digital form simultaneously to their physical production by the writer. The approach requires certain technological obstacles to be overcome (image capture speed, writing ergonomy and comfort) but has some significant potential applications, of which a few are listed here:

- Hand-written notes taken during a meeting can be captured in real time.
- Because the pointer movements are time-stamped, the dynamicity of the strokes can be recovered (in opposition to what happens when scanning a document *after* the writing has taken place). The time and order of each annotation can be preserved and related to other events, such as events in a video recording of a meeting (see [14, 23] for related ideas).
- The dynamicity of the strokes may help in signature authentification; it may also help in hand-writing recognition.
- Editor's marks can be hand-written on a draft typescript and interpreted in real-time for producing a corrected version.
- With a miniaturized wireless writer-pointer, notes taken on pads of Intelligent Paper (with no computer near-by) are subsequently available online. This could also be used to send hand-written faxes without a fax machine nearby.

[8] See section 5 for other ways of declaring peripherals by using Intelligent Paper.

Encryption Codes and Digital Commerce Another noteworthy extension of Intelligent Paper is obtained if the bits remaining, after *pointer-loc* and *page-id* data have been taken care of, are used for encryption purposes. The approach is the following. Suppose that at publication time, the publisher of an Intelligent Paper page associates with each individual *page-id* a certain string of, say, 50 bits. This encryption string *encrypt-string* is completely random and has no connection whatsoever with the *page-id* itself. It is printed on the code layer along with the *page-id* and *pointer-loc* codes. The relationship between the *page-id* and the *encrypt-string* is known only to the publisher.

The *encrypt-string* has applications to controlling access rights and to digital commerce. An example will suffice to give a gist of the approach. Consider the case of a recording company, *Digital Grammophon*, which distributes music recordings over the Net, for a certain fee. Let's now imagine the following scenario. *Digital Grammophon* sells, through record or book-shops, "listening cards" printed on Intelligent Paper and wrapped in a protecting envelope. One such card, for instance, lists the last six Mozart symphonies, in Bruno Walter's famous interpretation. Each movement is listed as a subheading under the corresponding symphony heading. The customer buys the card, goes back home, clicks on the heading for the first movement of Mozart's 40th, and the music is played for him on his Internet-enabled hi-fi system.

Digital Grammophon wants to prevent illegal use of the card. Each card has a unique, distinct, *page-id* and this already affords some protection to the company. For instance, if the card was copied by some means, *Digital Grammophon* could prevent the simultaneous use of the same *page-id* from two different users.[9] But the scheme is not completely air-tight. A skillful pirate, rather than simply copying the card, could try to discover how many "Bruno Walter's Mozart Symphonies" cards are sold by *Digital Grammophon* and what are their *page-ids* (in all probability, they would be consecutive numbers).

But, if, along with the *page-id*, the *encrypt-string* is coded on the card, *Digital Grammophon* can tell whether the *encrypt-string* is the right one for this *page-id*. Nobody else can guess the correct *encrypt-string* unless one is prepared to try the 2^{50} (roughly one million billion) possible combinations. Thus, the encryption code guarantees that although *copying is possible, forging is not*: the only way to obtain a record card is by duplicating an existing one, not by guessing a new one. It is, however, in the interest of the buyer of a card not to have it copied by other parties, unless it is for making a personal backup, which is acceptable, for very soon the countermeasures taken by *Digital Grammophon* would make his own card unusable.

Having discussed two *extensions* of the basic concept, we will now consider two variants of it which are rather *simplifications*.

[9] Several other schemes exist for controlling rights through the *page-id*. For instance, a given card could entitle its purchaser to a limited cumulative listening time.

Non-Positional Intelligent Paper We have mentioned above that invisible ink poses certain technological challenges, and that visible DataGlyphs could be used to provide some of the capabilities of Intelligent Paper. In all these cases, however, we have been considering a "positional" version of Intelligent Paper, that is, one in which the second level of indirection, from pointing events to actions defined relative to a digital page, is done via transmitting the (x, y) coordinates of the pointer relative to the physical page.

Another possibility exists. Suppose that, in a given multipage document, one has determined a certain number of "active areas", rectangular regions around words or icons in the document. Suppose now that the document is given a unique *document identifier*, which we write *doc-id*, and is similar to the *page-id* previously discussed.[10] Suppose also that each active area in the document is given a small numeric identifier *num-id* relative to this document. If a document contains no more than 1000 active regions, 10 bits are sufficient to uniquely identify each such region.

Now, it is possible to print, in each active region, a visible DataGlyph patch — which may be partially occluded by words or icons — that encodes the *num-id* for that region.[11] These patches, although visible, are visually non-obstructive and serve as a reminder to the user that a certain region is active. Suppose also that the *doc-id* is printed, again using visible DataGlyphs, in a cell at the bottom of each page (or possibly only once on the cover page of the document).

The user can then perform the following pointing actions. First he points to the *doc-id* at the bottom of the page to identify the digital-page (or rather the "digital-document") associated with the document. Then he points at some active region in the document. The pointer decodes the *num-id* of this region and transmits it to the digital page for interpretation, as in the scenarios discussed so far.[12]

Although this scenario loses several of the advantages of full positional Intelligent Paper — fully simulating a mouse, including ability to point to precise locations, to select a text or a graphical object or to draw — it still retains several important capabilities and is easier to implement in the short-term.

Intelligent Confetti and establishing links between Internet-embedded objects Intelligent Confetti are a further simplification of Intelligent Paper. They are small round stickers carrying a unique *page-id* code and nothing else. The globally unique *page-id* is transmitted to the *page-id* router which determines the address of the digital page associated with the confetti. In this way each confetti is globally associated with a unique digital document through which an action associated with the confetti can be performed. The crucial difference between

---

[10] In fact, even in the case of positional Intelligent Paper, it may be useful to define *doc-ids* for multipage documents.

[11] DataGlyphs, in some of their variants, can accept a relatively high-level of occlusion [9].

[12] The first step — pointing to the *doc-id* — need not be repeated each time, as long as the user is pointing in the same document.

confetti and conventional barcodes lies in the scheme for assigning a globally unique *page-id* to each confetti — this numbering scheme being under the control of the authorized Intelligent Paper producer — and for routing this *page-id* to its corresponding digital page in a universal way. Confetti can be used in several ways:

- Multimedia links can be added to conventional documents. By sticking a confetti in the margin of a conventional document, the user can associate a certain action with this confetti. The confetti producer provides the user with an interface on her computer through which she can define a link or an action to be associated with the confetti (for instance, ringing a phone number or playing a vocal annotation). When she later, from an arbitrary location, peruses the document again and points on the confetti, this action is performed for her.
- Confetti can be stuck on physical objects as well as paper documents. A confetti stuck on a mechanical part can later be used to display information about this part.

One of the most important uses of confetti is in establishing links between Internet-embedded objects. Until now, we have finessed the issue of how the "output peripheral close to the user" (see section 1) is actually identified to the Intelligent Paper communication infrastructure. One possibility is to associate the peripheral with the pointer through a conventional computer dialogue, where the pointer's owner is asked to identify a certain communication port at a certain Internet address, if the peripheral is a computer display or a printer, or to enter a certain telephone number if the peripheral is a telephone. With confetti, this operation can be done much more transparently: *the user simply clicks on a confetti stuck on the peripheral.* The action associated to this confetti consists in establishing the peripheral to the communication infrastructure as being the current peripheral, and in transmitting further information about the peripheral characteristics permitting the output action to be performed or optimized.

6 Conclusion

We have discussed some of the potential applications of Intelligent Paper, and presented the underlying technology in its broad outline. Some aspects of this technology — invisible ink under visible ink, writer-pointer — are not fully mature yet, but progress is being rapidly made. We often take for granted, and do not not even pause to consider, many technologies — color printing, miniaturized cameras, wireless telephony — which would be considered to be sheer science-fiction if it were not for the disturbing fact that they already exist.

Intelligent Paper is a proposal for making physical documents collaborate closely with digital documents. To take one instance where the tension between the two worlds is evident, several authors [16, 18] have opposed the practices of writing, publishing and reading books to their counterparts on the

Web. Whereas strengths of the book are: linearity, concentration of information, clear borderline between internal and external, uniform viewpoint maintained throughout, minimization of redundancy, careful choice of content, and publisher's reputation, those of the Web are: multidimensional connectedness, freedom of content, open-endedness, variety of viewpoint, unlimited content, and absence of juridiction over quality.

The very linearity of books imposes a lot of work on their designers. This effort is not simply an artifact of the book form, but an essential ingredient in the final result, which is organized for maximally efficient communication with the reader. The Web is more anarchic, more free, but the user needs to work out her own "design" for her reading.

Intelligent Books — books printed on Intelligent Paper — allow a complementarity between the two levels. The printed document corresponds to information which is highly salient, "obligatory" and controlled, the digital document and its links to information which is "on demand", open-ended, and interactive.

Whatever the current hype about Virtual Reality, the physical world is not going to disappear any time soon. The future of the human community is not about eliminating the feels and tastes of direct interaction with each other, places and objects, but about enriching this interaction. A world in which music stores, newsstands and bookshops have vanished would be a poorer world. A middle road will have to be found between digital information, dynamic, decentralized and open-ended but also somewhat cold and depersonalized and, on the other hand, traditional means of designing, producing and selling this information. We believe that Intelligent Paper represents one step towards such a balance.

Acknowledgments

Many thanks for their support and help to Hervé Gallaire, Monica Beltrametti, L. Noah Flores, Gregory Grefenstette, David Hecht, John Knights and Julian Reynolds.

References

1. Adobe Acrobat version 3.0, 1996. Adobe Systems Inc.
2. Graham G. Allan and Jean J. Robillard, editors. *Paper and the Electronic Age: Evolution or Revolution*, Guadalajara, Mexico, Oct 1996. SPIE — the International Society for Optical Engineering.
3. T. Arai, D. Aust, and S.E. Hudson. Paperlink: A technique for hyperlinking from real paper to electronic content. In *Proceedings of the ACM SigChi Conference on Human Factors in Computing Systems, CHI '97*. ACM, March 1997.
4. Thomas M. Destree, editor. *The Lithographers Manual*. Graphic Arts Technical Foundation, Pittsburgh, 1994.
5. James M. Duff, H. Bruce Goodbrand, and David L. Hecht. Authentication process. U.S. patent 05385803, 1995. Filed in 1993.

6. H. Bruce Goodbrand, James M. Duff, and Raymond W. Wong. Process for the authentication of documents utilizing encapsulated toners. U.S. patent 05208630, 1993. Filed in 1991.
7. David Hecht. Embedded data glyph technology for hardcopy digital documents. In *Proceedings of the SPIE Color Hard Copy and Graphic Arts III.* SPIE, Feb 1994.
8. David Hecht, 1997. Personal Communication.
9. David Hecht, L. Noah Flores, and Glen Petrie, 1997. Personal Communication.
10. Robert M. Hinden. IP next generation overview. *Communications of the ACM,* June 1996.
11. Joseph Jacobson, C. Turner, J. Albert, and P. Tsao. The last book. *IBM Systems Journal,* 36(3), 1997.
12. W. Johnson, S.K. Card, H. Jellinek, L. Klotz, and R. Rao. Bridging the paper and electronic worlds: The paper user interface. In *Proceedings of INTERCHI.* ACM, April 1993.
13. Michael G. Lamming. A Data Access System, 1992. European Patent Specification 0 495 612 B1, U.S. Patent 5535063.
14. Mik Lamming and William Newman. Activity-based information retrieval: technology in support of personal memory. In *Personal Computers and Intelligent Systems: Information Processing 92,* Amsterdam, North Holland, 1992.
15. William Newman and Pierre Wellner. A desk supporting computer-based interaction with paper documents. In *Proceedings of the ACM SigChi Conference on Human Factors in Computing Systems, CHI 92.* ACM, 1992.
16. Geoffrey Nunberg, editor. *The Future of the Book.* University of California Press, Berkeley, CA, 1996.
17. Kenton O'Hara and Abigail Sellen. A comparison of reading paper and on-line documents. In *Proceedings of the ACM SigChi Conference on Human Factors in Computing Systems, CHI '97.* ACM, March 1997.
18. Tim O'Reilly. Publishing Models for Internet Commerce. *Communications of the ACM,* June 1996.
19. Peter Robinson, Dan Sheppard, Richard Watts, Robert Harding, and Steve Lay. Animated paper documents. In *Proceedings of the 7th International Conference on Human-Computer Interaction,* San Francisco, August 1997.
20. Peter Robinson, Dan Sheppard, Richard Watts, Robert Harding, and Steve Lay. A framework for interacting with paper. In *Proceedings of Eurographics 1997,* Nov 1997.
21. Abigail Sellen and Richard Harper. Paper as an analytic resource for the design of new technologies. In *Proceedings of the ACM SigChi Conference on Human Factors in Computing Systems, CHI '97.* ACM, March 1997.
22. N. K. Sheridon, E. A. Richley, J. C. Mikkelsen, D. Tsuda, J. Crowley, K. Oraha, M. Howard, M. Rodkin, R. Swidler, and R. Sprague. The gyricon rotating ball display. In *Conference Record of the 1997 IDRC,* Toronto, Canada, Sept 1997. SID/IEEE.
23. Lisa Stifelman. Augmenting real-world objects: A paper-based audio notebook. In *Proceedings of the ACM SigChi Conference on Human Factors in Computing Systems, CHI '96.* ACM, April 1996.
24. Pierre D. Wellner. *Interacting with Paper on the DigitalDesk.* PhD thesis, University of Cambridge, Computer Laboratory, 1994.
25. Joseph Wright. Paper and the electronic age: Evolution or revolution. In Allan and Robillard [2].